Universidad de Alcalá
Escuela Politécnica Superior

Grado en Ingeniería en Tecnologías de Telecomunicación

Trabajo Fin de Grado

Analysis, implementation and evaluation of blind separation algorithms from audio sources

Autor: Rubén Centeno Álvarez

Tutores: Javier Macías Guarasa y Frank Sanabria Macías

2021

UNIVERSIDAD DE ALCALÁ

ESCUELA POLITÉCNICA SUPERIOR

Trabajo Fin de Grado

Analysis, implementation and evaluation of blind separation algorithms from audio sources

Autor: Rubén Centeno Álvarez

Directores: Javier Macías Guarasa y Frank Sanabria Macías

Tribunal:

Presidente: Juan Jesús García Domínguez

Vocal 1: Miguel González Herráez

Vocal 2: Javier Macías Guarasa

Calificación: .....

Fecha: .....

A mi familia y amigos

Acknowledgements

"Strength doesn't come from what you can do. It comes from overcoming the things you once thought you couldn't"

Rikki Roggers

Al llegar al final de esta etapa, probablemente una de las más importantes de mi vida, no puedo evitar mirar atrás en el tiempo y recordar lo que he vivido durante estos casi 5 años. Al hacerlo, me doy cuenta de que no podría haber llegado hasta aquí sin todas aquellas personas que me apoyaron y me dieron la fuerza necesaria para seguir adelante y conseguir lo que me propuse hace unos años, ser ingeniero. Por eso quiero agradecer a todos y cada uno de ellos el haber estado a mi lado durante todo el camino.

En primer lugar, gracias a mi familia por cada mensaje y llamada de apoyo, por preocuparse por mí e interesarse por lo que hacía, aún sin entender nada de lo que les contaba. Gracias especialmente a *la Besuguina y Andrea*, por ser mis mejores compañeras de aventuras, risas y locuras, por ser mis confidentes, por estar en los mejores momentos pero sobre todo por estar en los malos, en definitiva, por estar siempre ahí. Aunque si tengo que dar las gracias a alguien es a mis padres y a mi hermana. A los primeros, por sus consejos, por todos los valores que me han enseñado desde pequeño, por educarme en el esfuerzo y la dedicación, por hacerme saber que puedo tropezar y caer en el camino pero que lo más importante es siempre volver a levantarse, sois mi referente a seguir en la vida. Y *Nerea*, la pequeña de la casa, podría darte las gracias toda una vida y aun así no sería suficiente, gracias por cada risa, cada abrazo, cada vez que me has ayudado en una práctica sin saber muy bien lo que hacías, cada vez que me has dado fuerza y me has animado a continuar sin saberlo...gracias por todo.

En segundo lugar, quiero agradecer a los profesores que durante estos casi cinco años han conseguido transmitirme no sólo los conocimientos necesarios, sino también el amor por esta carrera. Me gustaría hacer una mención especial a Marta, Frank y Javi, que podría decir que han sido mis tres tutores. Gracias por estar siempre disponibles, por intentar siempre ayudarme a resolver cualquier duda que me surgía, por esos cafés que me hacían desconectar un rato, aunque fuera poco, de este proyecto final de carrera, y sobre todo por lo que he aprendido durante este último año.

Gracias a todos mis amigos, a los de la uni, especialmente a Laura y David, compañeros de interminables prácticas, de tardes en la biblioteca preparando exámenes, de cervezas, de muchos agobios, de grandes dudas existenciales, como *¿a=a?*, pero sobre todo de risas y buenos momentos. Sin vosotros la carrera no habría sido lo mismo. Gracias a los que se convirtieron en compañeros entre interminables horas de servicio diurno, y que luego se convertirían en amigos entre tardes de cervezas, noches de discotecas y muchas fiestas a las que lograron *liarme*. Gracias a los que muchas tardes y noches de *pizza* unieron de nuevo y a ese *Grupo Nuestro, Ostia*, que aunque nos vemos poco porque somos unos señores super ocupados, siempre me hacéis pasar tardes inmejorables. Gracias a todos por cada vez que habeis

conseguido alegrame el día y distraerme para olvidar los problemas, aunque fuera por un tiempo. Pero ante todo, gracias a Esther y Silvia, por ser las mejores amigas que se pueden tener, por escucharme, por animarme y celebrar cada pequeño logro como si fuera vuestro, por apoyarme en los malos momentos y sobre todo por *seguir siendo después de empezar a ser*.

Abstract

The purpose of this bachelor thesis work will be to perform a test bench with different methods used for Blind Source Separation (BSS) of people's voices, to be used in acoustic localization algorithms to improve their accuracy.

For this purpose, the MATLAB software tool will be used to implement and evaluate the different proposed systems. We will consider that most related systems in the scientific literature have three phases: move the audio source to the time-frequency (TF) domain by perfoming the Short Time Fourier Transform (STFT) to the audio mix, separate the audio into the different sources applying BSS techniques, and a final task with the obtained signals of reconstruction and transition to the time domain using the Inverse Short Time Fourier Transform (ISTFT).

For reaching the objectives described, the project has been divided in the following tasks: design of the blocks responsible for carrying out the STFT and the ISTFT that will be common for all BSS methods and the development of the BSS method bank and of the filtering stage, using a Wiener filter or similar, which will also be common to all methods. Last of all, testing and evaluation of the complete system using audio mixes obtained in similar environments to the one in which the system is to be applied to improve the location of the various sources.

Keywords: Blind Source Separation, Non-negative Matrix Factorization, Acoustic Source Localization, STFT, MATLAB.

Resumen

El propósito de este Trabajo de Fin de Grado (TFG) será realizar un banco de pruebas con distintos métodos utilizados para la separación ciega de fuentes de audio, Blind Source Separation (BSS), que será usado en algoritmos de localización acústica para mejorar su precisión.

Bajo este propósito, se usará la herramienta software MATLAB para implementar y evaluar los distintos sistemas propuestos. Consideraremos que la mayoría de los sistemas relacionados en la literatura científica se componen de tres fases: pasar al dominio tiempo-frecuencia mediante la realización de la Short Time Fourier Transform (STFT) de la mezcla de audio, separar la misma en las diferentes fuentes aplicando tecnicas Blind Source Separation (BSS), y la final reconstrucción de las señales obtenidas y paso al dominio del tiempo utilizando la Inverse Short Time Fourier Transform (ISTFT).

Con el fin de alcanzar los objetivos descritos, se ha dividido el trabajo en las siguientes tareas: diseño de los bloques encargados de realizar la STFT y la ISTFT, que serán comunes para todos los métodos BSS, desarrollo del banco de métodos BSS y de la etapa de filtrado, utilizando un filtro Wiener o similar, que también será común a todos los métodos. Finalmente, se probará y evaluará el sistema completo mediante mezclas de audio obtenidas en entornos similares al que se desea aplicar el sistema para mejoar la localización de las distintas fuentes.

Keywords: Separación ciega de fuentes, Factorización de matrices no negativas, Localizacion de fuentes de audio, STFT, MATLAB.

Extended Summary

The purpose of this bachelor thesis work will be to build a test bench with different methods used for Blind Source Separation (BSS) of people's voices, to be used in acoustic localization algorithms to improve their accuracy.

The work is centered on separating audio sources selected from different databases well known by the scientific community and with diverse algorithms, in order to test and validate some of them. These sequences are generated in different rooms and with several microphone array configurations, so that our implementation will address different scenarios. We will consider cases with a single person still in a room, others where a person is in continuous movement at different speeds, or even with a group of people who are talking at the same time.

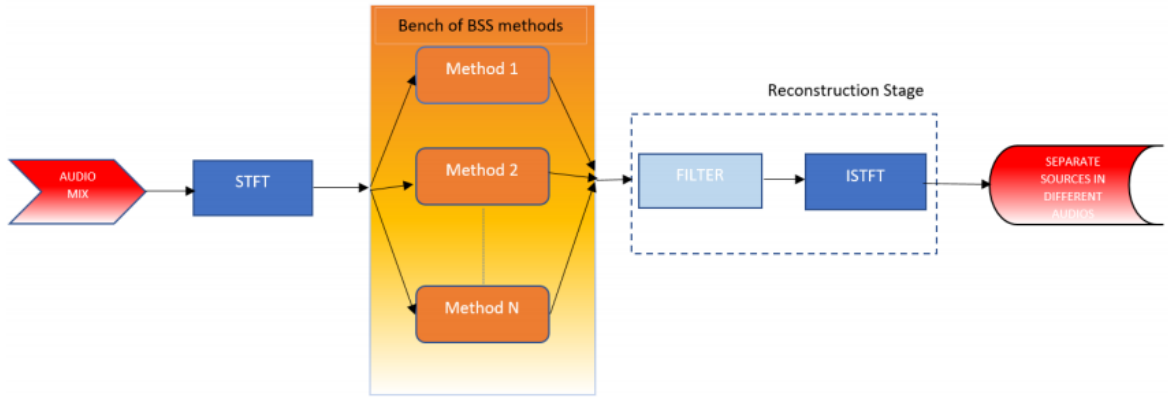


Figure 1: Prototype of the initial block diagram

The design of the system has been divided into different modules for ease of implementation, as detailed below and shown in Figure 1:

- A first block of code will be introduced to read from the databases the mix of audio that we want to separate into different sources.
- The STFT block will take care of transforming audio samples, thus moving from the time domain to the frequency domain, allowing us to have the signal described in each TF slot.
- A bank of BSS methods will then be implemented and tested with a switch, allowing to choose the method with which the test is being performed.
- A filter similar to the Wiener filter and a ISTFT block will be implemented in order to reconstruct the separated signal in the time domain.

The MATLAB software tool is used to carry out all these modules, which will also be used for the subsequent verification of the results.

We will start by defining a series of parameters that the system will use, such as the number of microphones, the size of the window used by the STFT, its displacement, the number of iterations, etc.

The STFT of the signals is then performed with the previously established parameters (thus obtaining the spectogram of the signal).Then, covariance matrix is computed, for each TF slot. We will assume that in each TF slot, each source behaves like a complex Gaussian variable. After that, the desired separation method can then be applied.

All this different BSS methods will be implemented and tested, with a particular focus on Non-Negative Matrix Factorization (NMF) due to its universality, flexibility to add limitations to the model and its good performance (see Figure 2). Although the basic techniques of NMF work only with non-negative values, we will use semi-NMF methods. This is done with the covariance matrix that models the phase difference between the signal arriving at each pair of microphones for every frequency value. This is a very useful information form one channel, since this phase difference information will give knowledge about the direction from which the sources arrive.

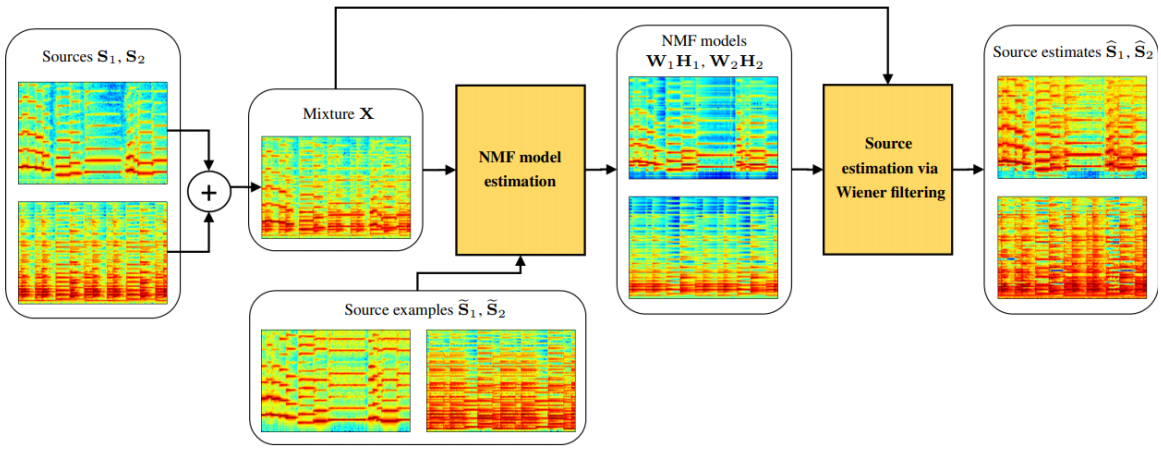


Figure 2: Figure from [1]. A general scheme of example-guided audio source separation based on NMF modeling.

The different methods are implemented on a bench and are:

- Ozerov.
- Sawada.
- Directional.

After going through the source separation algorithm, an estimation of the parameters that make up the different sources are obtained, that can be used to build a Wiener filter or similar to separate the sources.

Finally, the ISTFT of the reconstructed signals is performed to pass them to the time domain, so that they can be reproduced and the correct operation of the system can be checked.

Contents

List of Figures

List of Tables

List of Acronyms

AED	Adaptive eigenvalue decomposition.
ASL	Acoustic Source Localization.
BSS	Blind Source Separation.
DoA	Direction of Arrival.
GCC	Generalised cross-correlation.
ICA	Independent Component Analysis.
IS	Itakura-Saito.
ISTFT	Inverse Short Time Fourier Transform.
KL	Kullback-Leiber.
LGM	local Gaussian model.
MAP	maximum a posteriori.
ML	Maximum likelihood.
MM	majorisation-minimisation.
NMF	Non-Negative Matrix Factorization.
NTF	Non-Negative Tensor Factorization.
PCA	Principal Component Analysis.
STFT	Short Time Fourier Transform.
SVD	Singular Value Decomposition.
TF	time-frequency.
TFG	Trabajo de Fin de Grado.

Chapter 1

Introduction

1.1 Work context

Few would dispute the fact that, nowadays, one of the major lines of research is machine-human interaction. Within this field, one of the main tasks is to know the position of a person interacting within an environment. In this context, two methodologies appear, the invasive ones (in which it is necessary to wear portable devices to achieve localisation) and the non-invasive ones, which do not require this type of device and are therefore preferred over the former.

Two of the main techniques used in indoor location systems are those based on audio and video sensors. These modalities can be used separately or together to accomplish the task. Figure 1.1 shows a non-invasive localisation system based on audio and video sensors. In this work, the visual field will be left aside to focus on the auditory field. In this domain there are a large number of studies focused on obtaining the position of any acoustic source appearing in a scene from an array of microphones placed in the environment. This is known as Acoustic Source Localization (ASL). To achieve this, most methods attempt to estimate the Direction of Arrival (DoA) of sources, due to the simplicity and ease of access in many applications. These measurements can be obtained, for example, by using the Generalised cross-correlation (GCC) or the Adaptive eigenvalue decomposition (AED) algorithm.

Figure 1.1: Example of a multimodal localization system. Red circles shows where the microphone arrays are located

The aforementioned techniques have in common that they usually work with a set of microphone arrays distributed around the environment and the audio signal they capture. However, the aim of this thesis will be to test whether a prior separation of the different audio sources improves their localisation. This set of techniques is called BSS. Nowadays, there are many methodologies to achieve a correct source separation, Principal Component Analysis (PCA), Singular Value Decomposition (SVD), Independent Component Analysis (ICA) or Non-Negative Matrix Factorization (NMF) are some of them. The latter is the subject of interest of this paper due to its universality, its simplicity and the good results obtained in many studies.

In particular, this paper focuses on BSS using NMF methods. The main objectives of this thesis are detailed below.

1.2 Goals

This thesis is focused on the use BSS techniques on recordings of acoustic signals, occurring in a known environment, using an array of microphones.

The main purpose of this work is to develop a system that allows the separation of the different audio sources in a given sample, in order to check if the localisation of these sources is subsequently improved. To this end we propose the use of BSS techniques, more specifically NMF methods, which are implemented on a test bench. The inputs of our system are the raw audio signals from the set of microphones, and the outputs are the audios of the different sources separately. A simple schematic of the proposed system is shown in the following figure.

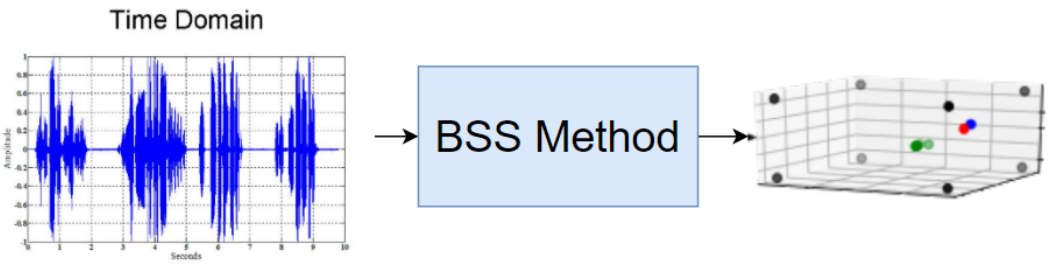


Figure 1.2: Schematic of the proposed system

To achieve this goal, two first blocks will be implemented to read the raw audio samples and to perform the STFT of them, thus passing to the frequency domain. Once in this domain, one of the NMF methods will be chosen from a set of them to achieve the separation of the different sources. Finally, a block will be implemented to reconstruct and return the signals to the time domain.

This system, as mentioned above, is implemented not with the objective of checking which one performs a better separation of the sources, but to evaluate which of them provides a better solution to improve the subsequent localisation of the different sources within the known environment.

1.3 Document organization

The rest of the document is organised in the following chapters:

- **State-of-the-art:** it begins with an overview of BSS techniques. It then introduces the NMF methods, explaining how they solve the problem of audio source separation and the different variants that have been implemented chronologically, with special emphasis on the methods that have subsequently been implemented in this work.
- **Proposed system:** this chapter details the architecture of the designed system and how the different separation methods have been implemented.
- **Experimental work and results:** shows the tests that have been carried out with different audio samples to check the correct functioning of the system.
- **Conclusion and future work:** this chapter contains the conclusions and coming research lines.
- **Appendixes:** the document includes two additional appendices that reflect the budget to carry out the work and the necessary software tools.

Chapter 2

State of the art

2.1 Introduction

We live in a world where we are surrounded by sound. Because of this, it is sometimes difficult to hear and to hold a conversation comfortably with the speaker we want to listen to. For this reason, it is important to be able to separate and extract the speech signal from the noise for both human-to-human and machine-to-human communications [2].

Audio source separation is an approach to estimate the source using information about the mixture observed in each input channel. This estimation is achieved by spatial filtering based on blind source localisation or time-frequency filtering based on audio source modelling, or both. [2]

Although until now Independent Component Analysis (ICA) has been used for audio source separation, because these signals in a real environment with reverberation are usually convoluted mixtures and are prevalent in many applications, their separation is a much more challenging task. Recently, NMF and DNNs have been exploited as well as other audio source separation techniques, with excellent results. Because of these good results, as indicated above, we will focus on NMF techniques [2].

NMF is a spectral decomposition technique in which, starting from a matrix $\mathbf{V}$ containing the input data, where the columns are the samples and the rows are the features, we try to find two unknown matrices by factoring them [2, 6].

$$\mathbf{V} \approx \hat{\mathbf{V}} \stackrel{def}{=} \mathbf{WH} \tag{2.1}$$

The matrices $\mathbf{W}$ and $\mathbf{H}$ function as a dictionary of recurring patterns and as the activation coefficients respectively. It is for this reason that we will henceforth refer to $\mathbf{W}$ as a dictionary and $\mathbf{H}$ as the activation matrix. The matrix $\mathbf{V}$ is of dimension FxN (where F denotes frequency and N denotes time), while the matrices $\mathbf{W}$ and $\mathbf{H}$ are of dimension FxK and KxN respectively, where K is usually the rank of the factorisation. These matrices, as the name of the technique itself indicates, will consist of non-negative factors [2].

Different techniques have been used over time to factor (2.1). The most notorious and oldest is Principal Component Analysis (PCA) which minimises the quadratic cost between $\mathbf{V}$ and its approximate $\mathbf{WH}$. The general principle of NMF-based audio spectral analysis is shown in figure 2.1 [2].

The figure 2.2 shows the result of applying NMF to a small piano sequence.

To achieve the factorisation we first solve the minimisation problem

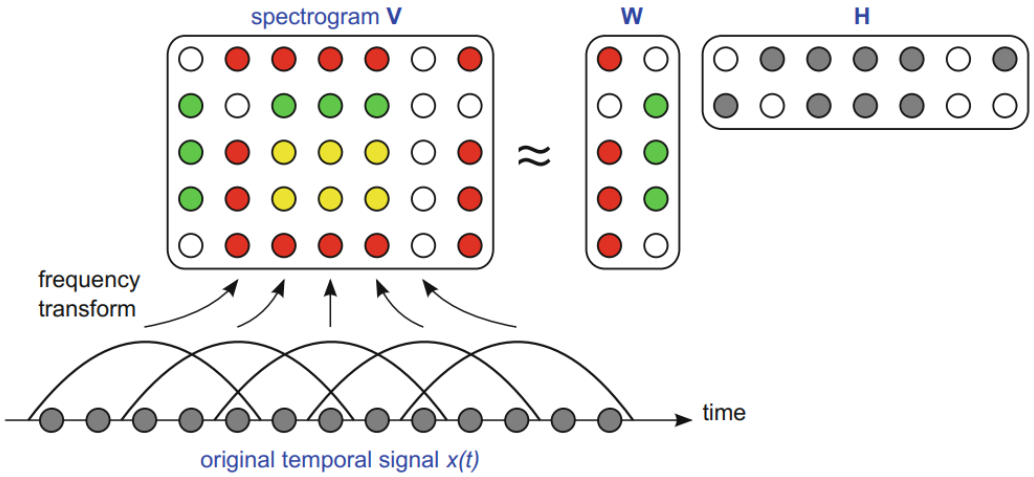


Figure 2.1: Figure from [2]. NMF-based audio spectral analysis. A short-time frequency transform, such as the magnitude or power short-time Fourier transform, is applied to the original time-domain signal $x(t)$. The resulting non-negative matrix is factorised into the non-negative matrices $\mathbf{W}$ and $\mathbf{H}$. In this schematic example, the red and green elementary spectra are unmixed and extracted into the dictionary matrix $\mathbf{W}$. The activation matrix $\mathbf{H}$ returns the mixing proportions of each time-frame (a column of $\mathbf{W}$)

$$\min_{\mathbf{W},\mathbf{H}} D(\mathbf{V}|\mathbf{WH}) \text{ subject to } \mathbf{W} \geq 0, \mathbf{H} \geq 0 \tag{2.2}$$

where $\mathbf{W} \geq 0$ and $\mathbf{H} \geq 0$ indicate the non-negativity of the matrix components, and $D(\mathbf{V}|\mathbf{WH})$ is a separable measure of fit such that

$$D(\mathbf{V}|\mathbf{WH}) = \sum_{f=1}^{F} \sum_{n=1}^{N} d([\mathbf{V}]_{fn}|[\mathbf{WH}]_{fn}) \tag{2.3}$$

where d is a positive cost function with a single minimum $x = y$ [2]. One of the most popular sets of NMF cost functions is beta-divergence, due to its success in audio signal processing. It can be defined as (2.4)

$$d_\beta(x|y) \stackrel{def}{=} \begin{cases} \frac{1}{\beta(\beta-1)}(x^\beta + (\beta-1)y^\beta - \beta\, x\, y^{\beta-1}), & \beta \in \mathbb{R} \setminus 0, 1 \\ x\, log\frac{x}{y} - x + y = d_{KL}(x|y), & \beta = 1 \\ \frac{x}{y} - log\frac{x}{y} - 1 = d_{IS}(x|y), & \beta = 0 \end{cases} \tag{2.4}$$

The limit cases $\beta = 0$ and $\beta = 1$ correspond to the Itakura-Saito (IS) and generalised Kullback-Leiber (KL) divergences, respectively. The case $\beta = 2$ corresponds to the quadratic cost $dQ(x|y)$ [2].

Factorisations with small positive values of β are relevant to decomposition of audio spectra, which typically exhibit exponential power decrease along frequency f [2].

The matrices $\mathbf{W}$ and $\mathbf{H}$ are successively updated until a stationary point is reached. However, because $C(\mathbf{W}, \mathbf{H})$ is jointly non-convex in $\mathbf{W}$ and $\mathbf{H}$, the stationary point may not be a global minimum (or even a local minimum). For this reason, in order to achieve a good factorisation, the initialisation of the parameters is very important, which is why it is often recommended to run the algorithm from different starting points [2].

$$\min_{\mathbf{W},\mathbf{H}} C(\mathbf{H}) \stackrel{def}{=} D(\mathbf{V}|\mathbf{WH}) \text{ subject to } \mathbf{H} \geq 0 \tag{2.5}$$

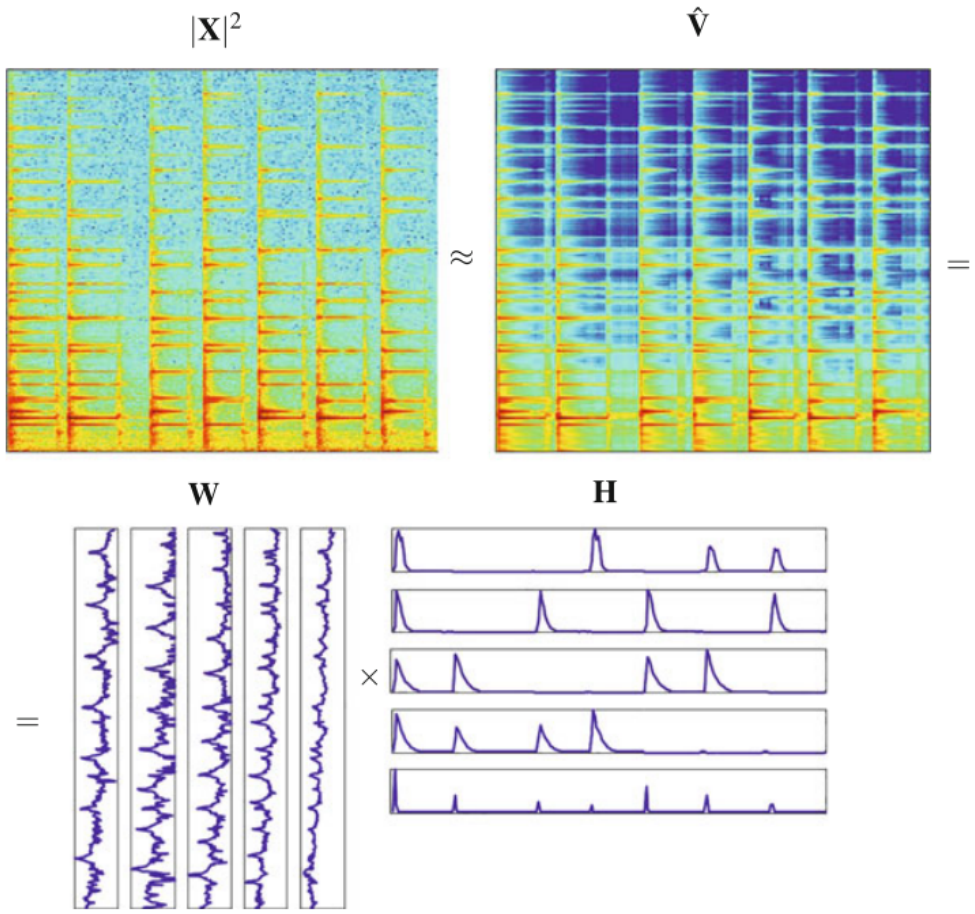


Figure 2.2: Figure from [2]. NMF applied to the spectrogram of a short piano sequence composed of four notes.

A common approach for conditional updates of $\mathbf{W}$ and $\mathbf{H}$ is majorisation-minimisation (MM). This technique consists of iteratively optimising an upper bound of the original objective function $C(\mathbf{H})$, which is easier to minimise [2].

This upper bound is achieved by decomposing $C(\mathbf{H})$ into the sum of a convex part and a concave part and upper bounding each of them separately. The convex part is upper bounded using Jensen's inequality and the concave part using the tangent inequality [2].

The resulting updates generalise can be written as

$$\mathbf{H} \leftarrow \mathbf{H} \circ \frac{\mathbf{W}^T((\mathbf{WH})^{\circ[\beta-2]} \circ \mathbf{V})}{\mathbf{W}^T(\mathbf{WH})^{\circ[\beta-1]}} \tag{2.6}$$

$$\mathbf{W} \leftarrow \mathbf{W} \circ \frac{((\mathbf{WH})^{\circ[\beta-2]} \circ \mathbf{V})\mathbf{H}^T}{(\mathbf{WH})^{\circ[\beta-1]}\mathbf{H}^T} \tag{2.7}$$

This update technique is called "multiplicative updates" and will be used by the methods described below [2] .

2.2 Ozerov

Multi-channel NMF can be formulated on the basis of a local Gaussian model (LGM) which is more general in itself (than multi-channel NMF) as it allows us to change spatial and spectral information in a systematic way [3].

$$\mathbf{x}_{fn} = \sum_{j=1}^{J} \mathbf{y}_{jfn} \tag{2.8}$$

Each image source is modelled as a complex circular Gaussian random vector of zero mean (equation 2.9) [3, 7]. Defined by two factors:

$$\mathbf{y}_{jfn} \sim N_c(0, \mathbf{R}_{jfn} v_{jfn}) \tag{2.9}$$

- Spatial covariance R_{jfn} representing the spatial characteristics of image source j at point TF
- Spectral variance v_{jfn} representing spectral characteristics

Once the spatial covariance and spectral variance are available, the random vectors $\mathbf{y}_{jfn}$ are assumed to be mutually independent in time, frequency and between sources [3] .

Given the audio mixing equation, and taking into account the independence discussed above, the STFT coefficients can be modelled as

$$x_{fn} \sim N_c(0, \sum_{j=1}^{J} \mathbf{R}_{jfn} v_{jfn}) \tag{2.10}$$

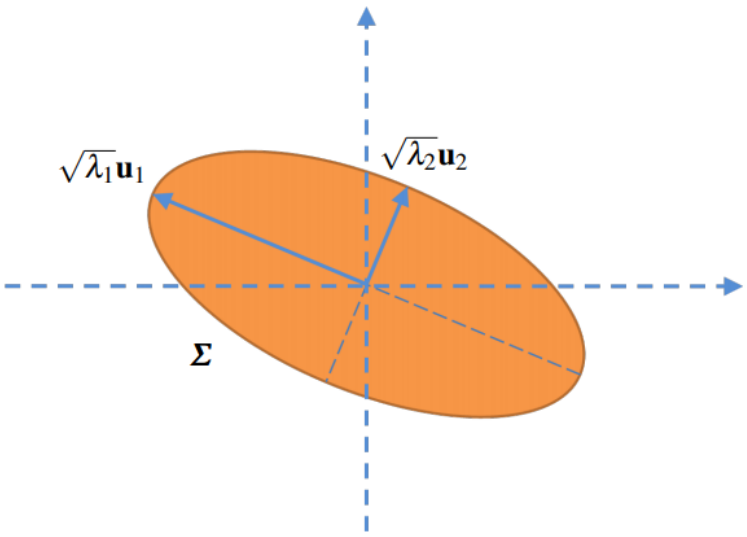


Figure 2.3: Figure from [3]. An illustration of a spatial covariance matrix $\mathbf{R}_{jfn}$ in the 2-channel case ($I = 2$). While dropping the indices j, f and n, the covariance matrix eigendecomposition may be written as $\mathbf{R} = \mathbf{U}\Lambda\mathbf{U}^H$, with being the eigenvectors and $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_1], \mathbf{u}_1, \mathbf{u}_2 \in \mathbb{C}^2$ being the eigenvectors and $\Lambda = diag([\lambda_1, \lambda_2]), \lambda_1, \lambda_2 \in \mathbb{R}_+$ being the eigenvalues. This illustration is not fully complete, since a 2D complex-valued covariance matrix is represented on a 2D real plane.

In our case the spectral variances are represented by lower rank matrices or tensors. Spectral covariance is not usually modelled with completely non-negative structures. For this reason we speak of a semi-negative model [3, 7].

For better understanding, an interpretation of the spatial covariance matrix is given and related to the methods used for multi-channel audio compression. In general this matrix is a full-rank positive definite Hermitian complex valued matrix. An example of such a matrix is depicted in figure 2.3 [3].

Because it is a complex-valued Hermitian, it can easily be shown that in the two-dimensional case it is encoded by only 4 real scalars [3].

2.2.1 NMF modeling of the sources

In this section the NMF spectral modelling of each source is presented. This is usually referred to as multi-channel NMF and consists of structuring the source variances in (2.9) with NMF structure as in the single-channel NMF case:

$$v_{jfn} = \sum_{k=1}^{K_j} w_{jfk} h_{jkn} \tag{2.11}$$

where the source-dependent $\mathbf{K}_j$ is usually smaller than both F and N, and w_{jfk} and h_{jkn} are all non-negative [3]. If we introduce non-negative matrices we can rewrite the equation (2.11) in a matrix form:

$$\mathbf{V}_j = \mathbf{W}_j \mathbf{H}_j \tag{2.12}$$

where $\mathbf{V}_j = [v_{jfn}]_{f,n} \in \mathbb{R}_+^{FxN}$, $\mathbf{W}_j = [w_{jfk}]_{f,k} \in \mathbb{R}_+^{FxK_j}$, and $\mathbf{H}_j = [h_{jkn}]_{k,n} \in \mathbb{R}_+^{K_j xN}$ [3].

A visualisation of this spectral NMF modelling is shown in the Figure 2.4

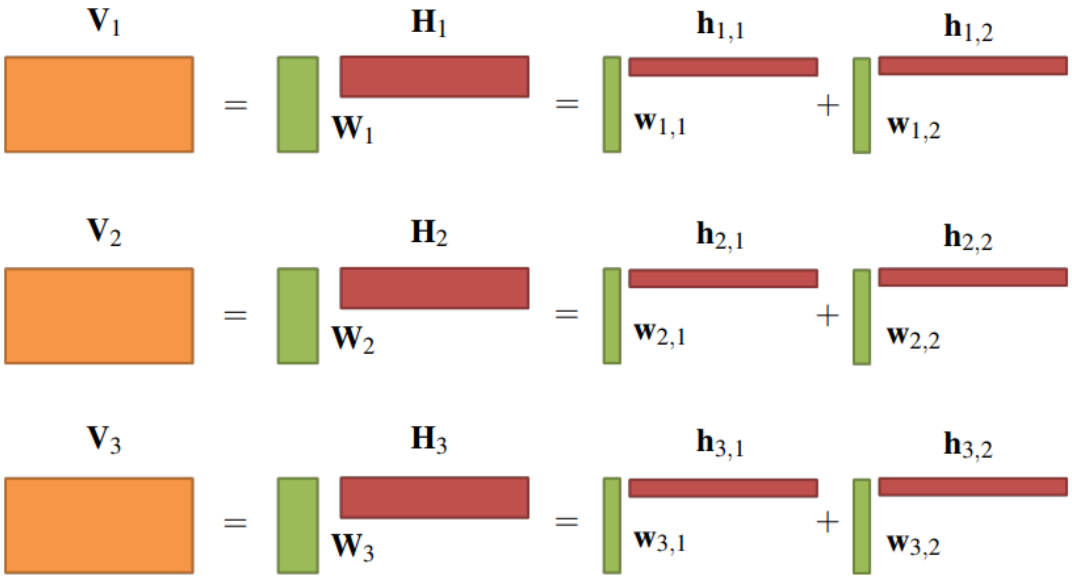


Figure 2.4: Figure from [3]. A visualization of spectral models of multichannel NMF. Source variances $\mathbf{V}_j$ of each of J (here $J = 3$) sources are modeled with NMF with K_j (here $K_j = 24$) components, which can be decomposed as a sum of K_j rank-1 matrices ($\mathbf{w}_{j,k}$ and $\mathbf{h}_{j,k}$ are the columns and the lines of matrices $\mathbf{w}$ and $\mathbf{H}$, respectively).

2.2.2 NTF modeling of the sources

While for the single-channel case it is necessary to fix the number of components K approximately or automatically calculated, which is not always easy, in the multi-channel case it is not only necessary to know the number of total components, but also the number of components of each source, which may vary from one source to another. To solve this problem, the following solution was proposed. Instead of representing each source as a single-channel NMF. All sources will share the NMF components. In addition, to specify the association between K components and J sources, a new non-negative matrix $\mathbf{Q} = [q_{jk}]_{j,k} \in \mathbb{R}_+^{JxK}$ is introduced. Each q represents the proportion of association of the K component with the J source [3].

$$v_{jfn} = \sum_{k=1}^{K} w_{fk} h_{kn} q_{jk} \tag{2.13}$$

where the columns of $\mathbf{Q}$ are normalized to sum to one, and each q_{jk} represent the proportion of association of the component k to the source j [3].

As already done in the NMF case, equation (2.13) can be rewritten in a tensor form:

$$\mathbf{V} = \sum_{k=1}^{K} \mathbf{w}_k \circ \mathbf{h}_k^T \circ \mathbf{q}_k \tag{2.14}$$

where "$\circ$" denotes the tensor outer product [3]. A visualization of Non-Negative Tensor Factorization (NTF) spectral models is shown on Figure 2.5

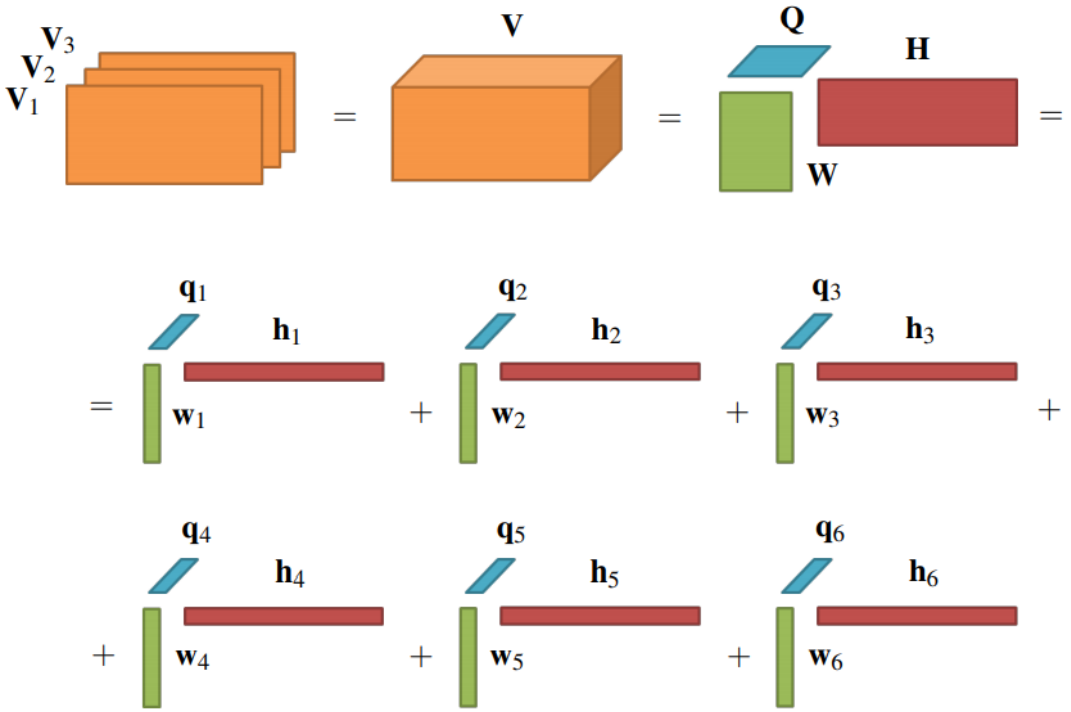


Figure 2.5: Figure from [3]. A visualization of spectral models of multichannel NTF. Source variances $\mathbf{V}_j$ are stuck in a common 3-valence tensor $\mathbf{V}$ modeled with PARAFAC model with K (here $K = 6$) components, which can be decomposed as a sum of K rank-1 3-valance tensors.

NTF modelling has several advantages over NMF modelling [3]:

- It is not necessary to specify the number of components at the outset for each source, but only the total number of components. This is because the components are placed automatically thanks to the **Q**-matrix.
- Some components are shared between different sources, making the modelling more compact.

Finally, it is added some restrictions on the spatial covariance to avoid overfitting. First of all, when the sources are static, the spatial covariances are assumed to be time invariant [3].

2.2.3 Model estimation criteria

Once the two proposed models have been presented, in order to estimate the model parameters from observed data, an estimation criterion is needed.

One of the most popular choices is the Maximum likelihood (ML) criterion that writes

$$\theta = arg\max_{\theta'} p(\mathbf{X}|\theta') \tag{2.15}$$

On the other hand, when an a priori distribution of the model parameters is specified, the maximum a posteriori (MAP) criterion is used instead of the ML criterion [3]:

$$\theta = arg\max_{\theta'} p(\theta'|\mathbf{X}) = arg\max_{\theta'} p(\mathbf{X}|\theta')p(\theta') \tag{2.16}$$

2.3 Sawada

This section includes explanations, algorithms and formulas made by Hiroshi Sawada to extend NMF to the multi-channel case.

In this section, as in the previous one, complex NMF is used to improve the capabilities for audio applications. Complex NMF takes into account phase information $x_{ij}/|x_{ij}|$, which is neglected in standard NMF. As we considered before, $x(t)$ is a microphone observation during a specific time to which the STFT is applied to obtain the complex matrix $\mathbf{X}$ whose elements are denoted as $x_{ij} \in \mathbb{R}$. Where i represents time and j represents frequency [8].

Sawada considers a stereo case, where matrices $X^{(1)}$ and $X^{(2)}$ are complex and represent the time-frequency domain observations of microphones 1 and 2. In this case, the multi-channel observation is modelled with the common non-negative matrices $\mathbf{T}$ and $\mathbf{V}$, plus a matrix explained below representing the inter-channel characteristics corresponding to the $\mathbf{T}$ matrix [8, 9].

Channel 1 is chosen as reference, where the inter-channel characteristics are normalised to 1. This modeling will look like:

$$p(\mathbf{X}^{(1)}|\mathbf{T},\mathbf{V},\mathbf{G}) = \prod_{i,j} N_c(x_{ij}^{(1)}|\sum_k g_{ijk}t_{ik}v_{kj}, 1) \tag{2.17}$$

While the second channel is modelled by introducing a matrix $\mathbf{H}$, the elements of which are complexes $h_{ik} \in \mathbb{C}$ and using the same $\mathbf{T}$, $\mathbf{V}$ and $\mathbf{G}$ matrices:

$$p(\mathbf{X}^{(2)}|\mathbf{T},\mathbf{V},\mathbf{G},\mathbf{H}) = \prod_{i,j} N_c(x_{ij}^{(2)}|\sum_k g_{ijk}h_{ik}t_{ik}v_{kj}, 1) \tag{2.18}$$

This matrix $\mathbf{H}$ contains information about the phase difference as well as the amplitude ratio between channels 1 and 2 [8].

In this case, we want to maximize the multichannel likelihood

$$p(\mathbf{X}^{(1)}, \mathbf{X}^{(2)}|\mathbf{T},\mathbf{V},\mathbf{G},\mathbf{H}) = p(\mathbf{X}^{(1)}|\mathbf{T},\mathbf{V},\mathbf{G}) \cdot p(\mathbf{X}^{(2)}|\mathbf{T},\mathbf{V},\mathbf{G},\mathbf{H}) \tag{2.19}$$

whose negative log-likelihood is given by

$$L_{mc}(\mathbf{T},\mathbf{V},\mathbf{G},\mathbf{H}) = \sum_{i,j}(|x_{ij}^{(1)} - \sum_k g_{ijk}t_{ik}v_{kj}|^2 + |x_{ij}^{(2)} - \sum_k g_{ijk}h_{ik}t_{ik}v_{kj}|^2) \tag{2.20}$$

An auxiliary function is defined in order to follow the iterative optimisation process

$$L_{mc}(\mathbf{T},\mathbf{V},\mathbf{G},\mathbf{H},S^{(1)},S^{(2)}) = \sum_{i,j}(\sum_k \frac{|s_{ijk}^{(1)} - g_{ijk}t_{ik}v_{kj}|^2}{r_{ijk}} + \sum_k \frac{|s_{ijk}^{(2)} - g_{ijk}h_{ik}t_{ik}v_{kj}|^2}{r_{ijk}}) \tag{2.21}$$

where $S^{(1)}$ and $S^{(2)}$ are newly tensors whose elements are complex and satisfy

$$\sum_k s_{ijk}^{(1)} = x_{ij}^{(1)} \ , \ \sum_k s_{ijk}^{(2)} = x_{ij}^{(2)} \tag{2.22}$$

and $r_{ijk}, \forall\, i,j,k$ are parameters satisfying $\sum_k r_{ijk} = 1$, $r_{ijk} \geq 0$ [8].

The minimization updates for this auxiliary function with respect to $S^{(1)}$ and $S^{(2)}$ are given by

$$s_{ijk}^{(1)} = g_{ijk}t_{ik}v_{kj} + r_{ijk}(x_{ij}^{(1)} - \sum_k g_{ijk}t_{ik}v_{kj}) \tag{2.23}$$

$$s_{ijk}^{(2)} = g_{ijk}h_{ik}t_{ik}v_{kj} + r_{ijk}(x_{ij}^{(2)} - \sum_k g_{ijk}h_{ik}t_{ik}v_{kj}) \tag{2.24}$$

And the minimization updates with respect to $\mathbf{T}$, $\mathbf{V}$, $\mathbf{H}$ and $\mathbf{G}$ are derived from its partial derivatives, and given by

$$t_{ik} = \frac{\sum_j \hat{x}_{ij}\Re[g_{ijk}^*(s_{ijk}^{(1)} + s_{ijk}^{(2)}h_{ik}^*)]}{(1+|h_{ik}|^2)\sum_j \hat{x}_{ij}v_{kj}} \tag{2.25}$$

$$v_{kj} = \frac{\sum_i \hat{x}_{ij}\Re[g_{ijk}^*(s_{ijk}^{(1)} + s_{ijk}^{(2)}h_{ik}^*)]}{\sum_i \hat{x}_{ij}t_{ik}(1+|h_{ik}|^2)} \tag{2.26}$$

$$h_{ik} = \frac{\sum_j \hat{x}_{ij}s_{ijk}^{(2)}g_{ijk}^*}{t_{ik}\sum_j \hat{x}_{ij}v_{kj}} \tag{2.27}$$

$$g_{ijk} = \frac{s_{ijk}^{(1)} + s_{ijk}^{(2)}h_{ik}^*}{|s_{ijk}^{(1)} + s_{ijk}^{(2)}h_{ik}^*|} \tag{2.28}$$

In summary, the negative log-likelihood is iteratively minimized by repeating (2.23)-(2.24) and one of the four updates (2.25)-(2.28).

This method does not work correctly, as **H** does not successfully reflect the inter-channel characteristics. This is because the complex phases have too much freedom to model the images accurately. To solve this problem there are two possible approaches [8].

- Impose a priori constraints to reduce the freedom, e.g. a spatial constraint on the excitation or a constraint on the clustering.
- Ignoring the phase g of each component t , and considering only the inter-channel features h. (Extension from standard NMF)

2.4 Directional

In contrast to the previous two methods, in this case, data on the directionality of sound propagation is used to improve NMF for source separation. Thanks to this information, the need for training data is eliminated, with the slight penalty of doubling the execution time. This new method was presented as the first to take advantage of the directionality of a set of microphones with a distance between them much smaller than the wavelength of the sound [4].

As introduced earlier, this method guides the NMF to find the different sources by providing estimates of the DoA obtained for each time-frequency slot from an array of microphones. This forms an **X** tensor of dimensions $frequencyxtimexdirection$, which shows the energy distribution in the environment. This jointly solves for all sources by finding the tensors **B** (direction distribution per source), **W** (spectral dictionary per source) and **H** (temporal performance of each element of the dictionary) to fit

$$X(f,t,d) \approx \sum_{s,z} B(d,s)W(f,z,s)H(t,z,s) \tag{2.29}$$

where s indexes the different sources and z indexes the sub-components of the dictionary.[4]

Among the main advantages obtained with this approach are the following [4]:

- Perceived separation quality - better than NMF;
- No supervision - no clean audio examples needed;
- Low overhead - computation on the same order as NMF;
- Suitability for small arrays - usable DoA estimates can be obtained from arrays which are much smaller than the wavelength of audible sounds and for which beamforming fails.

In this section we are going to use probabilistic language instead of matrix language as we have been doing in the previous sections. For this reason, rather than **X** we take a given probability distribution $p^{obs}(f,t)$ to decompose it as $p^{obs}(f,t) \approx \sum_z q(f,t,z)$, where q is factored as one of the following forms:

$$q(f,t,z) := q(f,z)q(t \mid z) = q(f \mid z)q(t,z) \tag{2.30}$$

as in Figure 2.6(a). The values of z index a dictionary of prototype spectra $q(f|z)$ which combine according to the time activations $q(t,z)$.[4]

To estimate the DoA in each time-frequency slot, the last squared method (probably the simplest) is used. The Fourier transform of the audio signals recorded by each of the M microphones is taken

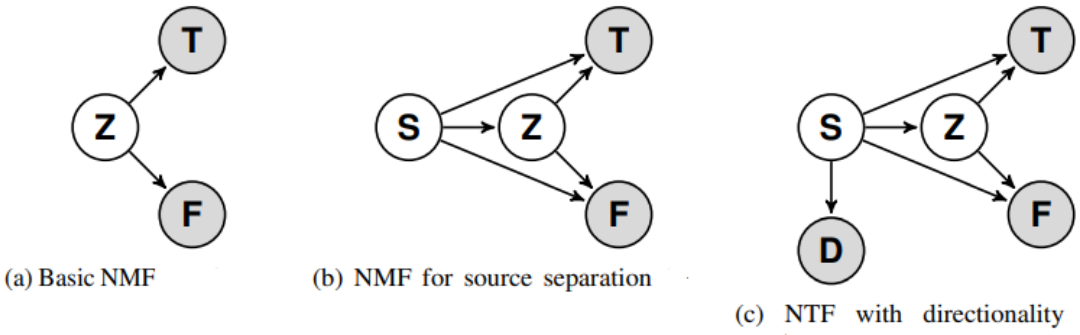


Figure 2.6: Figure from [4]. Graphical models for the factorizations

as given. This process is applied to all the slots, so we will focus on a single slot and its STFT value $Y_1, ..., Y_M$.[4]

It is assumed that this fringe is dominated by a point source far enough away to look like a plane wave and the array is small enough to avoid phase overlap $\angle Y_i$. If we denote x_i as the position of the microphone i and k as the wave vector, is obtained $\angle Y_i - \angle Y_1 = (x_i - x_1) * k$. To solve this linear equation for k, least squares will be applied. Serving k as an estimate of the DoA for the chosen slot. The different coefficients of k are fixed by the geometry, so it is possible to solve with a single pseudoinverse and a small matrix multiplication, the least squares problems for all TF bins [4].

In practice, instead of using a matrix $p^{obs}(f,t)$, we take as given a tensor (NTF) $p^{obs}(f,t,d)$, which will be interpreted as a distribution over time, frequency, and DoA quantified in a finite domain of size D. We decompose $p^{obs}(f,t,d) = p^{obs}(f,t)p^{obs}(d|f,t)$, where $p^{obs}(f,t)$ represents the normalised spectrogram and $p^{obs}(d|f,t)$ represents the estimate of the direction per TF slot. On the other hand, because the distance between the microphones is small, the variation of the signal amplitude between them is negligible, so we can derive the normalised spectrogram of one of the microphones [4].

We set $p^{obs}(f,t,d) \approx \sum_{s,z} q(f,t,d,z,s)$ for the factorization

$$q(f,t,d,z,s) := q(s)q(f|s,z)q(t,z|s)q(d|s) = q(d,s)q(f|s,z)q(t,z|s) \tag{2.31}$$

represented in Figure 2.6(c). A distribution $q(d|s)$ rather than a fixed DoA per source allows for noise, slight movements of sources, and modeling error [4].

The majorisation-minimisation method is used to fit (2.31) to $p^{obs}(f,t,d)$. This argument leads us to begin with a factored model:

$$q^0(f,t,d,z,s) := q^0(d|s)q^0(f|s,z)q^0(t,z|s), \tag{2.32}$$

forced the desired marginal to obtain

$$r(f,t,d,z,s) := p^{obs}(f,t,d)q^0(z,s|f,t,d), \tag{2.33}$$

and return to factored form by computing conditionals of r [4]:

$$q^1(d,s) := r(d,s), \quad q^1(f|s,z) := r(f|s,z), \quad q^1(t,z|s) = r(t,z|s). \tag{2.34}$$

2.5 Conclusions

In this chapter we have made a brief introduction to the subject of BSS, and then explained the NMF methods in more detail, explaining their theoretical basis, including equations and figures that describe how they work applied to single-channel cases. Next, the three NMF methods that have been implemented on the bench have been analysed in detail in order to deal with the multichannel-multilocutor case. Explaining how they are implemented and identifying their advantages over other different methods.

Chapter 3

Implementation

3.1 Introduction

This chapter of the book will detail the work carried out by the author. It will detail how the BSS bank responsible for separating the different audio sources that make up the mix has been implemented through programming. This code will be made up of codes that were previously made by the authors mentioned in the previous chapter (Ozerov, Sawada, etc.) and modifications that have been made to them to adapt them to our particular situation.

In order to facilitate the understanding of this system, pseudo-code will be included both for the general operation of the bank and for each of the methods that have been studied specifically.

Thus, the pseudocode representing the operation of the complete system divided into its different modules would look like 3.1

Algoritmo 3.1: Operation and different modules of the system

Result: Separates the different audio sources that make up the input mix
Deletion of possible variables stored in the system;
Introduction of the paths where the necessary libraries are located;
Input object creation;
Parameters definition;
STFT;
Apply separation algorithm;
Inverse STFT for each source;
Save separated speech on results folder;

3.2 Deletion of possible variables stored in the system

With this module, all possible variables stored in the workspace that could interfere with the correct functioning of the system are deleted. In addition, all the figure windows are closed and the command window is cleaned to facilitate the viewing of future results. The algorithm of this module is represented in 3.2

Algoritmo 3.2: Deletion module

Result: Delete all possible saved variables and close the open figure windows

```
close all;
clear;
clc;
```

3.3 Introduction of the paths where the necessary libraries are located

In this case, as the system needs, on the one hand, the different toolboxes made by the authors mentioned in previous sections, and on the other hand, the database containing the different audio mixes, it is necessary to add the paths where they are located so that the system can make use of them. To perform this task, it is simply necessary to use the algorithm 3.3

Algoritmo 3.3: Introduction of the paths

Result: Adds the specified folders to the top of the search path

```
addpath();
```

3.4 Input object creation

Once the necessary paths have been added, the next step is to create the audio mix to be fed into the system so that its different sources are separated. To do this, first the database to be used, in our case AV.16, must be selected. This database is presented in the results chapter, where the different sequences that compose it are detailed. After selecting the database, it is necessary to choose one of the available sequences, which will be the audio mix that will definitely enter the BSS system. All this is summed up in the algorithm 3.4

Algoritmo 3.4: Creation of the input audio mix

Result: Input object creation

```
Database selection;
Details of the sequences that make up the database;
Adding the database path;
Sequence selection;
```

3.5 Parameters definition

After having generated the audio mix to be fed into the system, it is necessary to set the value of some parameters in order for the system to function correctly.

First, it will be necessary to know the number of sources that make up the mix, the number of microphones in the room, and it will also be necessary to mark one of them as the reference microphone. On the other hand, it will be necessary to set the total number of NMF bases and the components for each source for the separation to work correctly.

Secondly, it is necessary to set the size of the STFT window and the size of its displacement, in order to achieve the highest possible resolution when passing the input audio to the frequency domain.

Finally, once all these parameters have been set, the user will be asked which of the three implemented BSS methods he/she wants to be used to achieve the separation of the different sources.

All these actions carried out in this module are detailed in the pseudocode 3.5

Algoritmo 3.5: Section of the code for setting the required parameter value

```
Result: Parameters configuration
Select the BSS method to be used;
Set number of sources;
Set the reference microphone;
Set fft size;
Set fft offset size;
Set number of total NMF bases;
Set number of iterations;
Set the number of microphones;
Set the number of NMF bases for each source;
Read the input audio mix;
```

3.6 STFT

In this section, after setting the configuration parameters and reading the audio signal, we proceed to the frequency domain and calculate its spectrogram using the STFT. To achieve this, the Sawada STFT function is used. This function must receive as input parameters

- The input audio signal, which has to have dimensions length x number of channels.
- Length of the frame (fftSize).
- Frame shift (shiftsize).
- Analysis window, which can be chosen from four possible windows: Hamming window, von Hann window, rectangular window, Blackman window and sine window

This function will first check if there is an error such as not enough elements are entered, the frame size is odd and will set some parameters to default values if they are not specified, such as the shift size (fftSize/2) or the analysis window (Hamming).

After checking that there is no error, zeros are added at the beginning and end of the signal before applying the FFT, thus achieving an interpolation of the samples in the frequency domain.

Finally, the STFT is calculated using the MATLAB fft function and the optimal analysis window function is calculated.

This whole process is synthesised in the pseudocode 3.6

3.7 Separation algorithms

Now that the spectrogram of the input audio signal has been obtained, we proceed to separate the different sources that make up the mix. To do this, a bank of three BSS methods has been designed by means of a switch, where each case represents one of the implemented methods.Once the program is run, it will ask the user which of the three methods he/she wants to use to perform the separation, and the user will indicate this via the command bar.

Algoritmo 3.6: Function that performs the STFT of the input audio

```
Result: Audio input in frequency domain
Function STFT(signal,fftSize,shiftSize,window) is
    Check errors and set default values;
    Pad zeros before and after signal values;
    Calculate STFT;
    Analysis window function used in STFT;
end
```

Algoritmo 3.7: Blind Source Separation Bank

```
Result: Audio source separation
switch method do
    case 1 (Ozerov) do
        Ozerov method;
    case 2 (Sawada) do
        Sawada method;
    end
    case 3 (Directional) do
        Directional method;
    end
end
```

The pseudo-code that represents this bank is 3.7.

In the following, we present how each of the methods has been implemented with their respective pseudocodes.

3.7.1 Ozerov

This section explains how the Ozerov method and its corresponding algorithm have been implemented. As can be seen in pseudocode 3.8, first the number of frequency and time slots are assigned, and the K value is calculated as a function of the components of each source and the number of sources. Subsequently, the matrices **A**, **W**, **H** and **Q** are initialised with random values. Finally, two functions are executed to separate and reconstruct the sources, which we will explain below.

Algoritmo 3.8: Ozerov method

```
Result: Audio source separation
The number of frequency slots is assigned;
The number of time slots is assigned;
Calculate K, K = nsrc * NMF_CompPerSrcNum;
A is randomly initialised;
W is randomly initialised;
H is randomly initialised;
Q is randomly initialised;
//NMF MU rules;
[Q_MU, W_MU, H_MU, cost] = multinmfconvmu(V, n_iter, Q, W, H, part, switch_Q, switch_W,
 switch_H);
//Reconstruction of the spatial source images;
Ie_MU = multinmf-recons-im(X, Q_MU, W_MU, H_MU, source-NMF-ind);
```

The function responsible for separating the different sources of the audio mix is contained in the code 3.9. This function will first check for errors and set some parameters to default values if they are

not specified. Next, the **V** and cost matrices are defined, and the value of the **V** matrix is calculated for each channel and each source. Finally, the values of the **Q**, **W** and **H** matrices are iterated *niter* times and scaled.

Algoritmo 3.9: Separation function

Result: Source separation
Function *multinmfconvmu(V, niter, Q, W, H, part, switchQ, switchW, switchH))* **is**
  If any arguments are missing, it initialises them to 1 by default;
  Definition of the matrices V_{ap} and cost;
  for $j \leq nsrc$ **do**
    for $i \leq nc$ **do**
      $V_{ap}(:,:,i) = V_{ap}(:,:,i) + repmat(Q(:,i,j),1,N).*P_j$;
    end
  end
  $cost(1) = sum(V(:)./V_{ap}(:) - log(V(:)./V_{ap}(:))) - F*N*n_c$;
  for $iter \leq niter$ **do**
    Update Q;
    Update W;
    Update H;
  end
  Scale Q/W;
  Scale W/H;
end

Algoritmo 3.10: Reconstruction function

Result: Reconstruction of the spatial source images
Function *multinmf-recons-im(X,Q,W,H,part)* **is**
  [F,N,n_c] = size(X);
  n_s = size(Q,3);
  P = zeros(F,N,n_s);
  for $j \leq n_s$ **do**
    P(:,:,j) = W(:,partj) * H(partj,:);
  end
  Im = zeros(F, N, n_s, n_c);
  for $i \leq nc$ **do**
    Prueba(:,:,:,i) = zeros(size(P));
  end
  for $j \leq ns$ **do**
    for $i \leq nc$ **do**
      Prueba(:,:,j,i) = repmat(Q(:,i,j),1,N) .* P(:,:,j);
    end
  end
  for $j \leq ns$ **do**
    for $i \leq nc$ **do**
      Im(:,:,j,i) = (Prueba(:,:,j,i) ./ sum(Prueba(:,:,:,i),3)) .* X(:,:,i);
    end
  end
end

Finally, once the different sources have been separated, they are reconstructed with code 3.10. This function initially defines the n_s and **P** matrices, and calculates the value of **P** for each source by mul-

tiplying the **W** and **H** matrices. Finally it is multiplied by the value of the **Q** matrix for each source and each channel, normalised with respect to itself and finally multiplied by the original input signal to achieve source separation.

3.7.2 Sawada

This section explains how the Sawada method and its corresponding algorithm have been implemented. As you can see in code 3.11, this method starts by calling a function which is explained below.

Algoritmo 3.11: Sawada method

```
Result: Audio source separation
[Y,cost] = bss-multichannelNMF(X,nsrc,nb,fftSize,shiftSize,it,drawConv);
```

This function in charge of font separation is shown in pseudocode 3.12. First, a delta value is set to avoid numerical computational instability. Subsequently, time-frequency-wise spatial covariance matrices are obtained and the observed spatial covariance matrix in each time-frequency slot is calculated. Finally, the sources are separated with the function detailed in pseudocode 3.13 and the signals are reconstructed using a Wiener filter.

Algoritmo 3.12: Bss-multichannelNMF Function

```
Result: Audio source separation
Function bss-multichannelNMF(X,ns,nb,fftSize,shiftSize,it,drawConv) is
    delta = 10^(-12);
    [I,J,M] = size(X);
    XX = zeros(I,J,M,M);
    x = permute(X,[3,1,2]);
    for i ≤ I do
        for j ≤ J do
            XX(i,j,:,:) = x(:,i,j)*x(:,i,j)' + eye(M)*delta;
        end
    end
    [Xhat,T,V,H,Z,cost] = multichannelNMF(XX,ns,nb,it,drawConv);
    Wiener filtering;
end
```

This function first checks if there is an error due to missing or erroneous arguments, and sets default values for those parameters that are not assigned one. Finally, the matrices **T**, **V**, **Z** and **H** are calculated by means of multiplicative updates.

Algoritmo 3.13: MultichannelNMF Function

```
Result: Audio source separation
Function multichannelNMF(X,N,K,maxIt,drawConv,T,V,H,Z) is
    Check errors and set default values;
    Xhat = local-Xhat( T, V, H, Z, I, J, M );
    Iterative update;
    Update T;
    Update V;
    Update Z;
    Update H;
end
```

3.7.3 Directional

Algorithm 3.14 explains how the Directional method works. In this case, first the coordinates of the microphones in the room are entered and the angles and directions of arrival of all signals at the microphones are calculated. Then, the variables are randomly initialised and the variables that will store the multiplicative update factor are created. Subsequently, q_{sft} is obtained by iterating $numIter$ times and for each source, summing over all z values. Finally, the different sources are reconstructed by multiplying the original input signal by q_{sft} for each source and each channel.

Algoritmo 3.14: Directional method

Result: Audio source separation
Entering microphone coordinates;
The arrival angles and directions of arrival of the radio signals at the microphones are calculated;
q_{fsz} is initialised randomly and normalised to the sum of all elements;
q_{tsz} is initialised randomly and normalised to the sum in z;
q_{ds} randomly initialised and normalised with respect to the sum in d;
$q_{deltadft}$ is initialised;
Variables storing the multiplicative update factor are created;
for $iter \leq numIter$ **do**
  Calculation of q_{dfts} ;
  Plot q_{ds} ;
  for $src \leq nsrc$ **do**
    Sum over all z values for each source to get q_{fts};
  end
end
We finally have what we wanted (q_{sft}): an F x T x S tensor with the fraction of each source in each time-frequency bin;
//Signal reconstruction;
for $ch \leq numCh$ **do**
  for $src \leq nsrc$ **do**
    $Y(:,:,src,ch) = q_{sft}(:,:,src). * X(:,:,ch)$;
  end
end

3.8 ISTFT

After having achieved the separation and reconstruction of the signals from the different sources that made up the audio mix, we proceed to transfer these signals to the time domain and their subsequent graphical representation in different figures. To achieve this, the Sawada ISTFT function is used. This function must receive as input parameters

- Spectrogram of the input signal.
- Frame shift (shiftsize).
- Analysis window, which can be chosen from four possible windows: Hamming window, von Hann window, rectangular window, Blackman window and sine window.
- Length of the original signal.

As with the STFT function, this function will first check for any errors such as not enough elements being entered, the number of rows not being odd, and will set some parameters to default values if they are not specified, such as the Hamming window.

After checking that there are no errors, the optimal synthesis window is calculated based on the minimum distortion principle and the ISTFT is calculated using the MATLAB ifft function.

Finally, discarding padded zeros in the end of the signal.

This whole process is synthesised in the pseudocode 3.15

Algoritmo 3.15: Function that performs the ISTFT of the separate sources

```
Result: Separate sources in the time domain
for src ≤ nsrc do
    sep(:,:,src)=Function ISTF(spectrogram, shiftSize, window, orgLength) is
        Check errors and set default values;
        Calculate optimal synthesis window based on minimal distortion principle;
        Inverse STFT;
        Discarding padded zeros in the end of the signal;
    end
    Plot of the separate audio sources in the time domain;
end
```

3.9 Saving of separate sources

Finally, after having all the sources separated in the time domain, we will proceed to save the results using the code 3.16.

Algoritmo 3.16: Saving of separate sources

```
Result: Creation of the different .wav files containing the different sources separated and
        received by the different microphones
for iSource ≤ nsrc do
    y = sep(:,:,iSource);
    for iMic ≤ nMics do
        audiowrite
    end
end
```

3.10 Localisation block

Once the different sources that make up the mixture have been separated, they are fed into the localisation block to estimate the position of the sources. The algorithm starts by loading the parameters of the experiment, such as the number of microphones, the method used or the database. It then creates a network of 3D points around the ground truth in order to locate the speaker. Then, the localisation process begins, starting with the reading of the audio signal, to later perform a pre-processing and remove zero mean, pre-emphasis and windowing, SRP-PHAT is calculated and the maximum SRP-location is obtained. Finally, the error between the estimated position and the true position of the source is evaluated, representing graphically this difference of positions in millimetres, and a document with the estimated coordinates of the speakers is generated.

3.11 Conclusions

In this chapter, the pseudocode of the different blocks that make up the system has been compiled. It began by indicating the structure of the complete system, mentioning the different modules that make it up. Then, in each section, the functioning of each of the blocks has been detailed, explaining and adding the code of the most important functions that make up each of them. In this way, not only the code that had already been implemented by each of the authors mentioned in the text is included, but also the code that has been developed and included in this thesis in order to adapt the code base to our test case.

Chapter 4

Results

4.1 Introduction

This chapter will show the results obtained at the end of the work, as well as the database that has been used. It will be divided into 4 sections. Section 4.2 describes the database used, giving details of the scenario where it was recorded and the different sequences. Section 4.3 contains the results of applying two specific sequences from this database to our system, and section 4.4 groups together the conclusions we draw from the results obtained.

4.2 Database

The database used for system evaluation is AV16.3 [5], recorded in the *Smart Meeting Room* of the IDIAP research institute, which consists of a $8.2m \times 3.6m \times 2.4m$ rectangular room containing a centrally located $4.8m \times 1.2m$ rectangular table.

This database has a variety of different scenarios, from "meeting situations" where the speakers are seated most of the time, to "motion situations" where the speakers are moving most of the time.

The location of the speakers is specified in an L-shaped area around the table in the room, as shown in Figure 4.1. This area in which the speakers can move has a length of 3 metres and a width of 2 metres, minus 0.6 metres of width which is occupied by the table [5].

Three cameras and two 8-microphone arrays of 10 cm radius are used, which are 0.8 m apart. The position of the cameras is optimised, using a loop process that includes the calibration of the cameras with 2D information only. On the other hand, each camera is calibrated using information from the 2D and 3D measurements in the reference of the microphone arrays.

The reason why all this hardware is used is:

- Recording with several cameras generates different points of view from which people can be observed.
- Recording with two array microphones creates test cases for 3D audio source location and tracking, as each array of microphones can provide an estimate of the location of each audio source [5].
- In order to compute the 3D coordinates of an object from the 2D coordinates of the image planes of the cameras, at least two cameras must be used [5] .

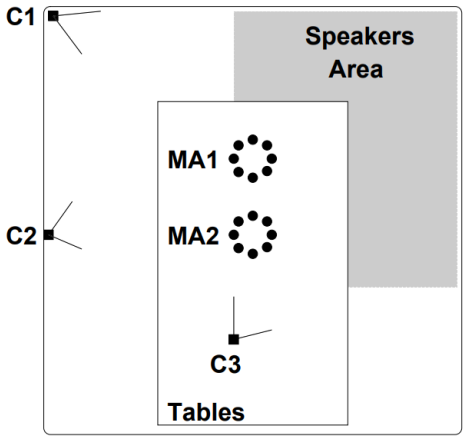


Figure 4.1: Figure from [5].Physical setup: three cameras C1, C2 and C3 and two 8-microphone circular arrays MA1 and MA2. The gray is in the field of view of all three cameras. The L-shaped area is a 3 m-long by 2 m-wide rectangle, minus a 0.6 m-wide portion taken by the tables.

So, the database contains audio and video data taken by the 3 video cameras, and the two circular microphone arrays. The cameras have a frame rate of 25 f.p.s ($40ms$ period) while the audio has been recorded at 16 kHz. The dataset is fully labeled, providing the mouth ground truth location. Synchronization information between the audio and video streams is also available.

In the experiments we have used a total of 8 annotated sequences, listed in table 4.1.

Sequence name	Duration (seconds)	Modalities of interest	Nb. of speakers	Speaker(s) behavior	Desired annotation
seq01-1p-0000	217	A	1	S	M, seg
seq11-1p-0100	30	A, V, AV	1	D	M, F, seg
seq15-1p-0100	35	AV	1	S,D(U)	M, F, seg
seq18-2p-0101	56	A(ov)	2	S,D	M, seg
seq24-2p-0111	48	A(ov), V(occ)	2	D	M, F
seq37-3p-0001	511	A(ov)	3	S	M, seg
seq40-3p-0111	50	A(ov), AV	3	S,D	M, F
seq45-3p-1111	43	A(ov), V(occ), AV	3	D(U)	H

Table 4.1: Table from [5]. List of the annotated sequences. Tags mean: [A]udio, [V]ideo, presominant [ov]erlapped speech, at least one visual [occ]lusion, [S]tatic speakers, [D]ynamic speakers, [U]nconstrained motion, [M]outh, [F]ace, [H]ead, speech/silence [seg]mentation

- **seq01-1p-0000** A single speaker, static while speaking, at each of 16 locations covering the shaded area in Fig 2. The speaker is facing the microphone arrays. The purpose of this sequence is to evaluate audio source localization on a single speaker case [5].
- **seq08-1p-0100** One speaker, mostly moving while speaking. Always facing the cameras and microphones. The speaker is talking most of the time [5].
- **seq11-1p-0100** One speaker, mostly moving while speaking. The only constraint on the speaker's motion is to face the microphone arrays. The speaker is talking most of the time [5].
- **seq15-1p-0100** One moving speaker, walking around while alternating speech and long silences [5].

- **seq18-2p-0101** Two speakers, speaking and facing the microphone arrays all the time, slowly getting as close as possible to each other, then slowly parting [5].
- **seq24-2p-0111** Two moving speakers, crossing the field of view twice and occluding each other twice. The two speakers are talking most of the time [5].
- **seq37-3p-0001** Three speakers, static while speaking. Two speakers remain seated all the time and the third one is standing. Overall five locations are covered. Most of the time 2 or 3 speakers are speaking concurrently [5].
- **seq40-3p-0111** Three speakers, two seated and one standing, all speaking continuously, facing the arrays, the standing speaker walks back and forth once behind the seated speakers [5].
- **seq45-3p-1111** Three moving speakers, entering and leaving the scene, all speaking continuously, occluding each other many times. Speakers' motion is unconstrained. This is a very difficult case of overlapped speech and visual occlusions [5].

4.3 Results and discussion

The results of the experiments that have been carried out are given below. In this section only localisation results will be presented and not separation results. This is because we don't have access to the true separated sources, so that we cannot evaluate by objective metrics how good is the separation we have achieved from our estimated sources. We can only subjectively evaluate the separation by listening to the audios and observing the time (or frequency) domain representation of the audios (as in Figures 4.2 and 4.3), which do not provide sufficient clarity nor objective information on the separation performance.

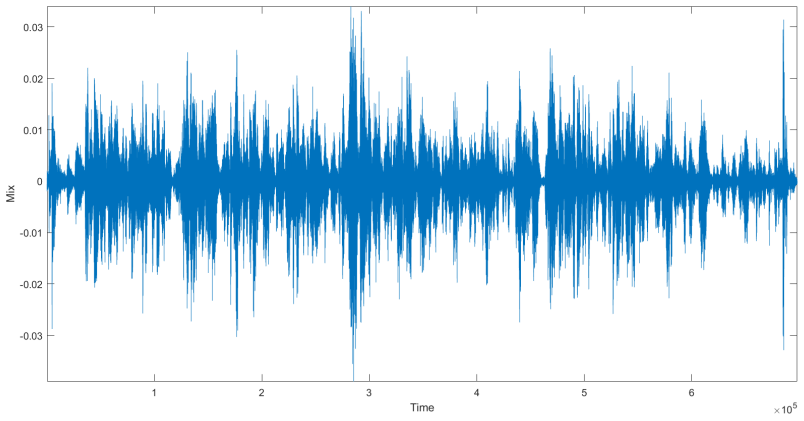


Figure 4.2: Audio mix of three sources.

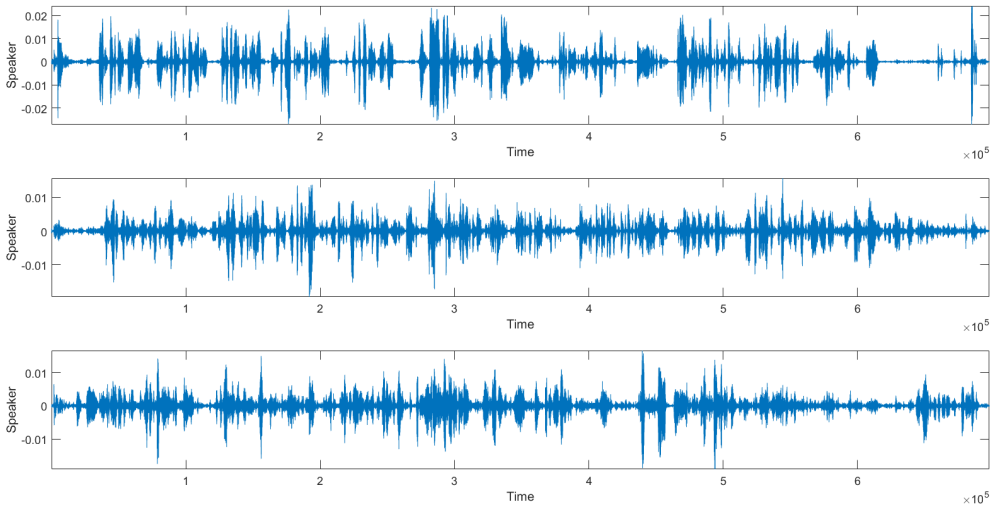


Figure 4.3: Three separate speakers.

It is for this reason that we will only evaluate the quality of the location of the speakers after the separation. For this purpose, the system will be run with two different audio samples from the database. The first will be sequence **seq08-1p-0100**, which, as indicated in the previous section, consists of a man in continuous movement around the room who keeps talking almost all the time. Although when we do source separation we expect there to be more than one speaker, in this case we are going to apply our bank of methods to an audio mix composed of a single person to try to separate the background noise from the speaker's voice. So in this case the system will be configured to detect 2 different sources, trying to separate the voice from the noise. Secondly, sequence **seq18-2p-0101** will be used, which involves 2 seated speakers talking at the same time. This sequence is used to test the correct functioning of the NMF methods to separate two overlapping voices. On the other hand, the values of different variables will be varied to check how they affect localisation. In this case, the number of NMF bases that make up the audio, the size of the STFT window and the size of its displacement will be varied. In order to test which of the three methods results in better localisation, all will be simulated with the same configuration sets.

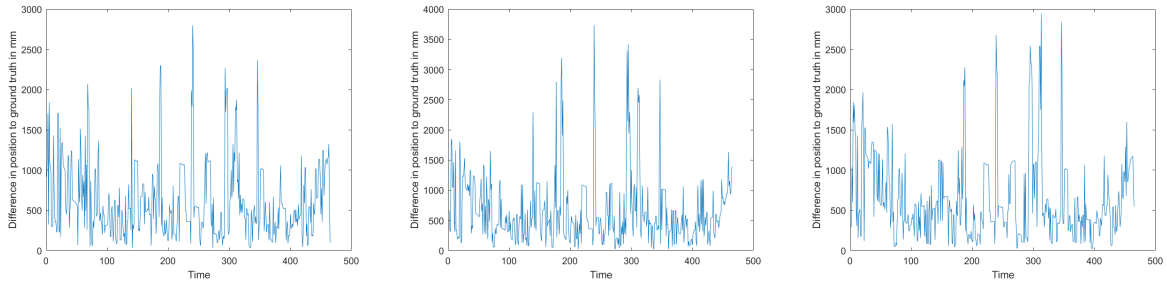
4.3.1 Audio with a single speaker (seq08-1p-0100)

In this first experiment, as explained above, this sequence will be used to try to separate the background noise from the locutor's voice, and thus try to check if the speaker's localisation is improved thanks to a cleaner signal compared to the case in which the background noise is not eliminated.

4.3.1.1 Ozerov

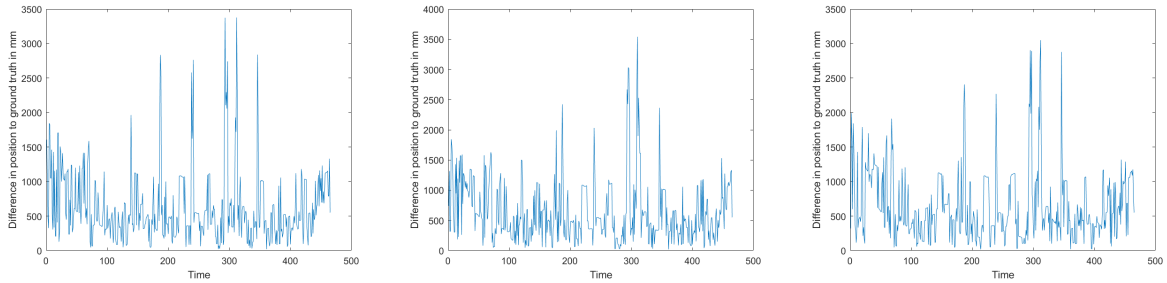
This method, as well as the other two methods, will be simulated by initially setting a maximum of 8 NMF bases for the whole mixture and gradually changing the STFT window size and shift. Giving values of 1024/512, 2048/1024 and 4096/2048 respectively as can be seen in Figures 4.4a,4.4b and 4.4c.

Subsequently, the three methods will be simulated again, with the same three window sizes but in this case setting 16 NMF bases instead of 8 as set out in Figures 4.5a,4.5b and 4.5c.



(a) Window size/shift 1024/512. (b) Window size/shift 2048/1024. (c) Window size/shift 4096/2048.

Figure 4.4: Localisation result applying Ozerov's method with 8 NMF bases, and different window sizes/shifts.

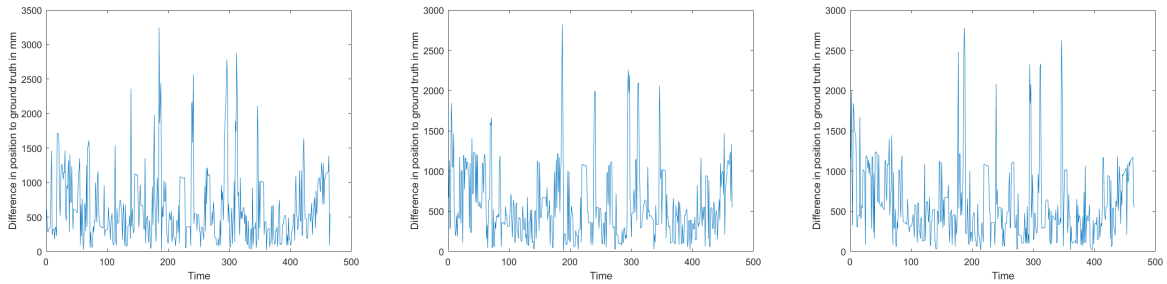


(a) Window size/shift 1024/512. (b) Window size/shift 2048/1024. (c) Window size/shift 4096/2048.

Figure 4.5: Localisation result applying Ozerov's method with 16 NMF bases, and different window sizes/shifts.

4.3.1.2 Sawada

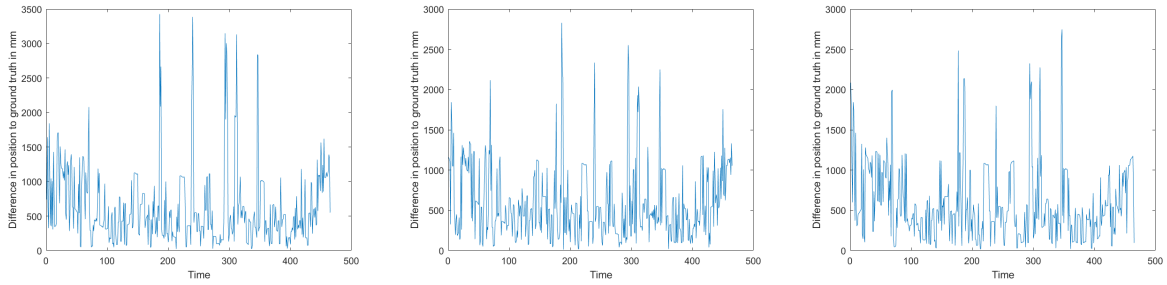
In this section we will perform the same experiments as in the previous section, except that this time the Sawada method will be used. First, a value of 8 NMF bases will be set, thus obtaining the results shown in the Figures 4.6a, 4.6b and 4.6c.



(a) Window size/shift 1024/shift. (b) Window size/shift 2048/shift. (c) Window size/shift 4096/shift.

Figure 4.6: Localisation result applying Sawada's method with 8 NMF bases, and different window sizes/shifts.

Subsequently, the three methods will be simulated again, with the same three window sizes but in this case setting 16 NMF bases instead of 8 as set out in Figures 4.7a,4.7b and 4.7c.

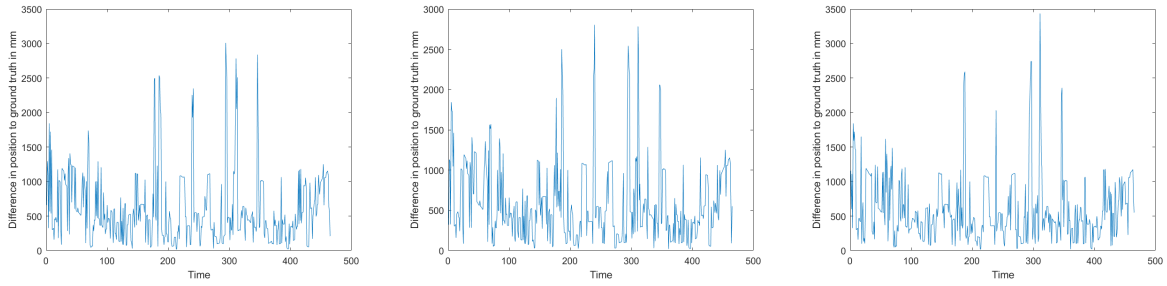


(a) Window size/shift 1024/shift. (b) Window size/shift 2048/shift. (c) Window size/shift 4096/shift.

Figure 4.7: Localisation result applying Sawada's method with 16 NMF bases, and different window sizes/shifts.

4.3.1.3 Directional

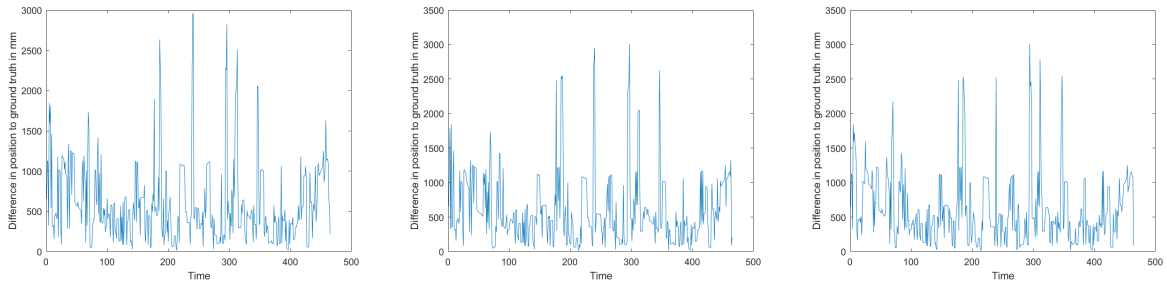
In this section we will perform the same experiments as in the previous section, except that this time the directional method will be used. First, a value of 8 NMF bases will be set, thus obtaining the results shown in the Figures 4.8a, 4.8b and 4.8c.



(a) Window size/shift 1024/512. (b) Window size/shift 2048/1024. (c) Window size/shift 4096/2048.

Figure 4.8: Localisation result applying the directional method with 8 NMF bases, and different window sizes/shifts.

Subsequently, the three methods will be simulated again, with the same three window sizes but in this case setting 16 NMF bases instead of 8 as set out in Figures 4.9a,4.9b and 4.9c.



(a) Window size/shift 1024/512. (b) Window size/shift 2048/1024. (c) Window size/shift 4096/2048.

Figure 4.9: Localisation result applying the directional method with 16 NMF bases, and different window sizes/shifts.

As can be seen from the results obtained for the three methods, we can affirm that the localisation of the speaker is not sufficiently correct. Despite the fact that after listening to the resulting audios, in most cases the speaker's voice was correctly separated from the noise, although in some points the difference in

distance between the estimated position and the real one is less than 30 cm, in many others it is almost equal to 3 metres. This could be due to the fact that by using only audio to achieve localisation, when there are moments of silence or other noises, this could be seriously affected. As, for example, this causes silent frames to exist and therefore the error may increase, which could be solved with a voice activity detector..

4.3.1.4 Summary results

The results in the figures above are meant to provide a general picture of the precision we can achieve with each of the methods (with errors up to almost 3 meters), but a clearer quantitative metric is required to assess which of them is actually better than the others and which are the factors affecting this performance.

To do so, we provide Table 4.2, in which the average error in millimeters is shown, for each of the evaluated methods and control parameters.

		Size/Shift		
Method	NMF basis	1024/512	2048/1024	4096/2048
Ozerov	8	616mm	650mm	633mm
	16	629mm	647mm	640mm
Sawada	8	630mm	574mm	582mm
	16	629mm	596mm	583mm
Directional	8	601mm	589mm	588mm
	16	594mm	598mm	606mm

Table 4.2: Mean error in millimetres between estimated position and ground truth with sequence (seq08-1p-0100)

The results are similar among the different methods, but the directional method seems to provide the lowest average error in most of the cases.

We can also observe that when we increase the number of total NMF bases, the localisation result worsens, as well as when we increase the size of the window and its displacement, except in the case of applying Sawada's method, where if we increase the size of the window, the error is reduced.

Spectrograms in Figures 4.10 and 4.11 are then shown below to give a visual idea on the quality of the audio separation. In Figure 4.11, it can be seen how the original audio seems to be separated into voice and noise, as opposed to the Ozerov method which seems to separate high and low frequencies, as can be seen in Figure 4.10, thus concluding that the task we had of separating the ambient noise from the voice, with the directional method seems to be possible and that may be the reason why the error in the subsequent localisation is lower.

Additional experimentation and a detailed error analysis is needed to provide statistically significant results and conclusions.

4.3.2 Audio with two speakers (seq18-2p-0101)

As has been shown in the previous subsection, the use of BSS methods does not introduce any improvement in the location of the speakers. This can be seen from the fact that there are several points where the difference in distance between the estimated position and the true position is up to 3 metres. Based on this premise, we now present the results obtained when there are two speakers instead of one in the audio

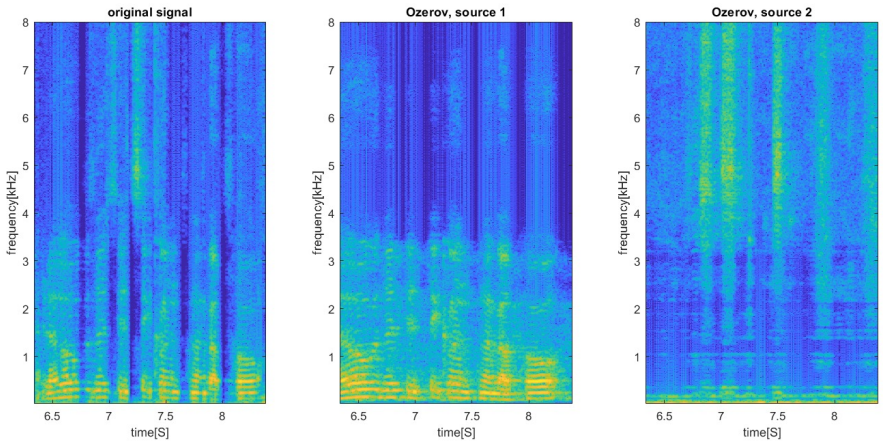


Figure 4.10: Spectrogram of a section of the original sequence (seq08-1p-0100) and of the two separated sources by the Ozerov method.

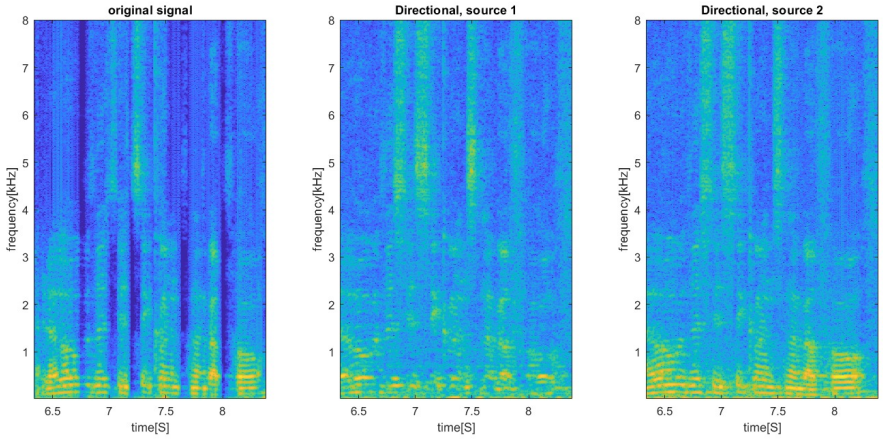
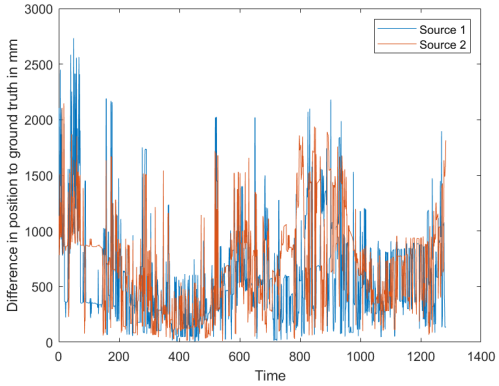


Figure 4.11: Spectogram of a section of the original sequence (seq08-1p-0100) and of the two sources separated by the directional method.

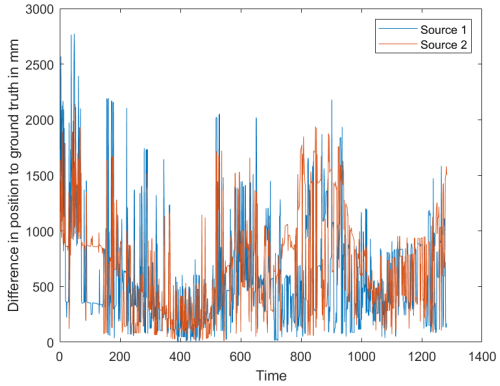
mix, assuming that since this task is more complicated than the previous one, where only the ambient noise was separated from the speaker, the results obtained will also be poor in terms of localisation.

In this case, the three implemented methods were tested again, this time setting the number of NMF bases to 12 (4 per source, as before) and only with two window sizes, 1024 and 4096, in order o better appreciate the differences. As the results obtained are similar to those of the previous section and hardly differ from each other, it has been decided to include only the Figures 4.12a, 4.12b and 4.12c, so as not to overload the document with almost identical information.

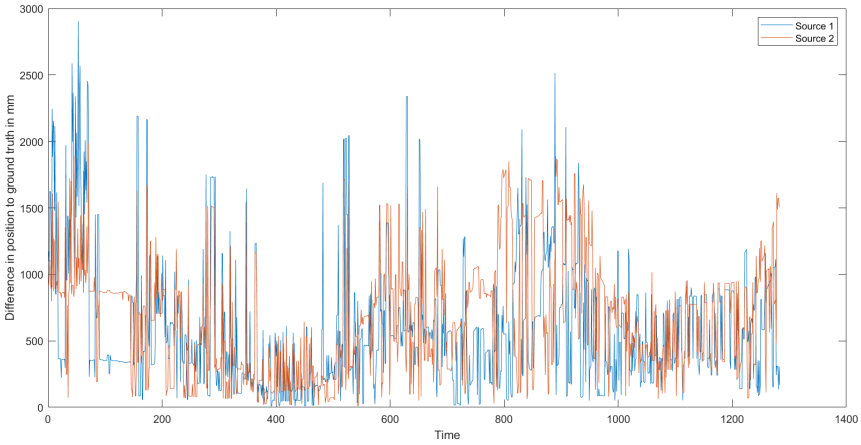
Again, as can be seen in the last three figures, we can state that the localisation achieved after separating the different audio sources is not accurate. This is because at many points the positional error is again above two metres. However, in this case the result could have been more expected, as now when listening to the audio of the estimated sources, it could be seen at several points how the voices of the two speakers overlapped and a correct separation was not achieved. So it was almost certain that the localisation was going to fail as well. Moreover, if we start from the fact that the single-speaker case is supposed to be easier to separate and localise, when this one failed, we could sense that the multi-speaker case was not going to have good results either. This is confirmed by looking at the figures, where we can see that many points are more than 30 centimetres apart, and therefore we can affirm that the localisation



(a) Sawada, window size/shift 4096/2048.



(b) Ozerov, window size/shift 4096/2048.



(c) Directional, window size/shift 4096/2048.

Figure 4.12: Localisation result applying Ozerov's and directional methods in multispeaker mixes with 12 NMF bases, and different window sizes/shifts.

is not correct. This may again be due to losing track of the speakers when they are silent, or to losing track of them when both speakers are too close to each other.

As we did for the case of the single speaker task, we provide a summary performance metric in 4.3, showing the average localization error for this task. Again, the directional method is the one showing the lowest error, closely followed by the other two.

We do not show the summary results for different window sizes, but the conclusions are similar to those discussed in the previous section.

The spectrogram of the sequence with two speakers appying the Ozerov method is shown in Figure 4.13, where it can be seen that it mixes information from the two speakers, resulting in two audios in which the two voices are heard overlapping at all times, one in the background of the other. This may be the reason why the localisation error in this case is so high compared to the rest.

Again, additional experimentation and a detailed error analysis is needed to provide statistically significant results and conclusions.

			Shift/Size
Method	NMF basis	Speaker	4096/2048
Ozerov	12	1	600
		2	684
Sawada	12	1	592
		2	687
Directional	12	1	585
		2	677

Table 4.3: Mean error in millimetres between estimated position and ground truth with sequence (seq18-2p-0100)

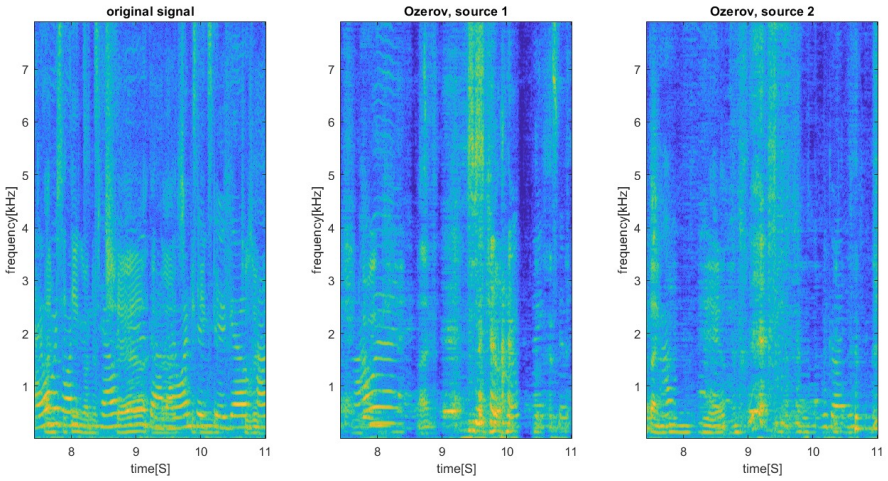


Figure 4.13: Spectrogram of a section of the original sequence (seq18-2p-0100) and of the two separated sources by the Ozerov method.

4.4 Conclusions

This chapter of the book contains the results obtained after carrying out the relevant experiments, as well as information on the database used. The number of sequences used has been indicated and the reason for this selection has been explained. The reason why the quality of the separation algorithms cannot be assessed has been explained and the quality of the localisation has been directly assessed. During the chapter, graphs representing the difference in mm spacing between the estimated position at each point and the ground truth have been included, and the average localization error results have been summarized in specific tables.

The main conclusions are that the directional method seems to be the one to achieve the best results in both the single and multiple speaker case, with higher errors when increasing the number of NMF basis and window sizes.

We could not run additional experimentation nor a detailed error analysis to further obtain relevant insights in the operation of the different methods, so this is the most important task to be addressed in future work.

Chapter 5

Conclusions and future work

5.1 Conclusion

The objective of this thesis was to create a bank with different BSS methods that would allow us to analyse whether a previous separation of the different sources that make up an audio mix would give us better results when trying to locate the speakers than by applying only the SRP-based localisation algorithm.

To achieve the bank design, the MATLAB software tool was used to design the skeleton of the bank and the stages prior to separation. On the other hand, for the separation stage, we have used the algorithm already implemented by the authors mentioned in the paper, adapting them to our multichannel case study.

From the obtained results, the methods implemented in the bank do not perform a good enough separation for our case study (multi-channel and multi-speaker), and the subsequent localisation of the sources gets very high localization errors, thus concluding that the results are not good enough to be used in the localization systems being developed in our group. Certainly, additional work must be carried out to do a more thorough and extensive evaluation.

5.2 Future works

The following future lines derived from this work are proposed:

- Carry out a more detailed study on the operation and performance of the proposed methods, using additional testing sequences, as time limitations did not allow to do so during the development of this work
- Use another existing database or record our own database consisting of audio sequences more in line with the implemented BSS methods. This could lead to better results in the separation of the different sources and therefore a possible improvement of the subsequent localisation.
- As a result of having our own database, from which we would have access to the separated real sources. Metrics could be implemented to assess whether the separation of such sources from an audio mix is good or not.

- Implement different NMF methods in the BSS method bank. One of the methods that could be included is GCC, as recent studies have shown that, after applying a localisation method based on SRP after having separated the different sources using GCC, good localisation results are achieved.
- Use techniques other than NMF methods, such as Principal Component Analysis (PCA), Independent Component Analysis (ICA) or Singular Value Decomposition (SVD). In order to see if these methods yield better results than NMF.
- Include a voice activity detector so that audio localisation is performed only when speakers are speaking, in order to reduce error due to stages of silence.

Bibliography

[1] A. Ozerov, S. Kitić, and P. Pérez, "A Comparative Study of Example-guided Audio Source Separation Approaches Based On Nonnegative Matrix Factorization," in *MLSP 2017 - Machine Learning for Signal Processing*, Tokyo, Japan, Sep. 2017. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01578378

[2] S. Makino, *Audio source separation*, 8 2018.

[3] A. Ozerov and C. Févotte, "Multichannel nonnegative matrix factorization in convolutive mixtures for audio source separation," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 18, pp. 550 – 563, 04 2010.

[4] N. D. Stein, "Nonnegative tensor factorization for directional blind audio source separation," *ArXiv*, vol. abs/1411.5010, 2014.

[5] G. Lathoud, J.-M. Odobez, and D. Gatica-Perez, "Av16.3: An audio-visual corpus for speaker localization and tracking," in *Machine Learning for Multimodal Interaction*, S. Bengio and H. Bourlard, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 182–195.

[6] L. Girin, S. Gannot, and X. li, *Audio source separation into the wild*, 11 2018.

[7] A. Ozerov and C. Févotte, "Multichannel nonnegative matrix factorization in convolutive mixtures for audio source separation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, pp. 550–563, 2010.

[8] H. Sawada, H. Kameoka, S. Araki, and N. Ueda, "Formulations and algorithms for multichannel complex nmf," *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 229–232, 2011.

[9] ——, "Multichannel extensions of non-negative matrix factorization with complex-valued data," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, pp. 971–982, 2013.

Appendix A

Solicitation document

The elements necessary to carry out the project are specifically detailed below. Indicating the hardware and software used

A.1 Physical elements

- ASUS ZenBook laptop, for system programming.

A.2 Software

- Data processing and analysis: Matlab, versión 2020b, toolboxes .
- Flowcharts made with: https://app.diagrams.net/.
- Text editor: TeXstudio (Latex).

Appendix B

Budget

In this last chapter an estimate of the total cost to realise the project is included. Then, the costs are presented divided according to their origin, showing the subtotal in each of the subsections, and at the end the total of these figures is shown.

B.1 Software resources

In order to carry out this project, it is necessary to use the Matlab software tool, with which the system responsible for separating the different audio sources has been implemented.

The table B.1 details the cost of the licence for this application.

Concept	Unit price	Coefficient	Subtotal
Matlab	200,00 €	0,0833	16,66 €

Table B.1: Software resources

B.2 Human resources

The human resource cost comes from the manpower of an engineer who is responsible for the development of the entire system. This cost is included below.

Concept	Hourly rate	Number of hours	Subtotal
Engineer	50,00 €	500	25000,00 €
TOTAL			25000,00 €

Table B.2: Human resources

B.3 Material execution budget

The material execution budget is the sum of the software and human resources necessary to carry out the work.

Concept	Subtotal
Software resources	16,66 €
Human resources	25000,00 €
TOTAL	25016,66 €

Table B.3: Material execution budget

B.4 Amount of the contract execution

The costs of execution by contract include the costs arising from the use of the facilities where the work has been carried out, tax charges, financial expenses, administrative fees and project control obligations.

This expenditure is assumed by establishing a surcharge on the cost of the amount of the material execution budget. This surcharge is equivalent to 22% of this amount.

Concept	Subtotal
22% of the total cost of material execution	5503,67 €

Table B.4: Amount of the contract execution

B.5 Facultative fees

A percentage of 7% of the total cost of execution by contract is fixed for this project.

Concept	Subtotal
7% of the contract execution cost	385,26 €

Table B.5: Facultative fees

B.6 Total budget

The table B.6 shows the sum of all the budget items taken into account in the previous sections.

Concept	Subtotal
Material execution budget	25016,66 €
Amount of the contract execution	5503,67 €
Facultative fees	385,26 €
TOTAL (without IVA)	30905,59 €
IVA (22 %)	6799,23€
TOTAL	37704,82 €

Table B.6: Total project budget

Universidad de Alcalá
Escuela Politécnica Superior

ESCUELA POLITECNICA
SUPERIOR